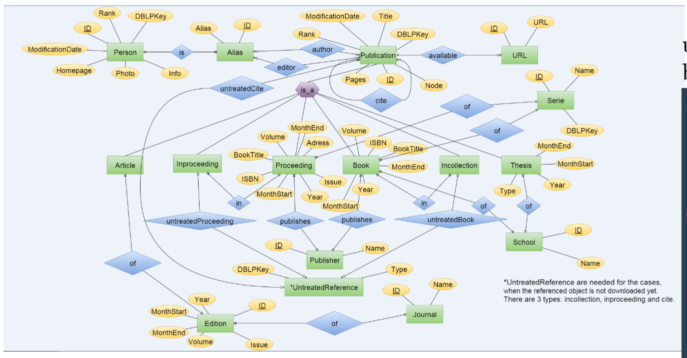
Project report

Sergey Solonets
Innopolis University
420500 Innopolis
Innopolis, Russia
ssolonets@gmail.com

Kozar Anastasia
Innopolis University
420500 Innopolis
Innopolis, Russia
nastya.shev9594@gmail.com

1. ER-DIAGRAM



*UntreatedReference are needed for the cases, when the referenced object is not downloaded yet. There are 3 types: incollection, inproceeding and cite.

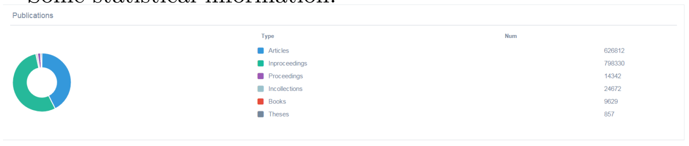
2. SITE DESCRIPTION

Structure of our site:

2.1 Home

2.1.1 *Dashboard*

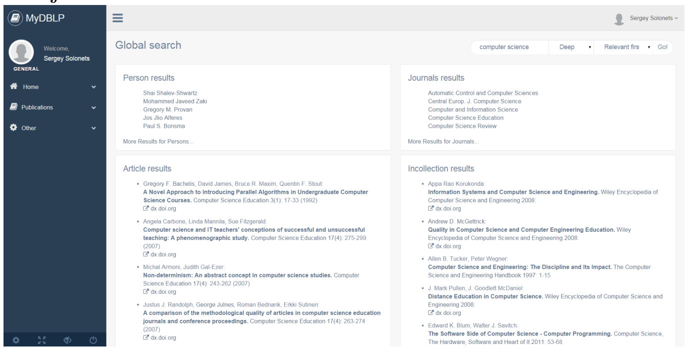
Some statistical information.



Type	Num
🟦 Articles	626812
🟩 Inproceedings	798330
🟪 Proceedings	14342
▨ Incollections	24672
🟥 Books	9629
⬛ Theses	857

2.1.2 *Global search*

Allows to search through all types of publications, authors and journals.



Global search

Person results

Shai Shalev-Shwartz
Mohammed Javeed Zaki
Gregory M. Provan
Jos Jlio Alferes
Paul S. Bonsma

More Results for Persons...

Journals results

Automatic Control and Computer Sciences
Central Europ. J. Computer Science
Computer and Information Science
Computer Science Education
Computer Science Review

More Results for Journals...

Article results

- Gregory F. Bachelis, David James, Bruce R. Maxim, Quentin F. Stout:
  A Novel Approach to Introducing Parallel Algorithms in Undergraduate Computer Science Courses. Computer Science Education 3(1): 17-33 (1992)
  dx.doi.org
- Angela Carbone, Linda Mannila, Sue Fitzgerald:
  Computer science and IT teachers' conceptions of successful and unsuccessful teaching: A phenomenographic study. Computer Science Education 17(4): 275-299 (2007)
  dx.doi.org
- Michal Armoni, Judith Gal-Ezer:
  Non-determinism: An abstract concept in computer science studies. Computer Science Education 17(4): 243-262 (2007)
  dx.doi.org
- Justus J. Randolph, George Julnes, Roman Bednarik, Erkki Sutinen:
  A comparison of the methodological quality of articles in computer science education journals and conference proceedings. Computer Science Education 17(4): 263-274 (2007)
  dx.doi.org

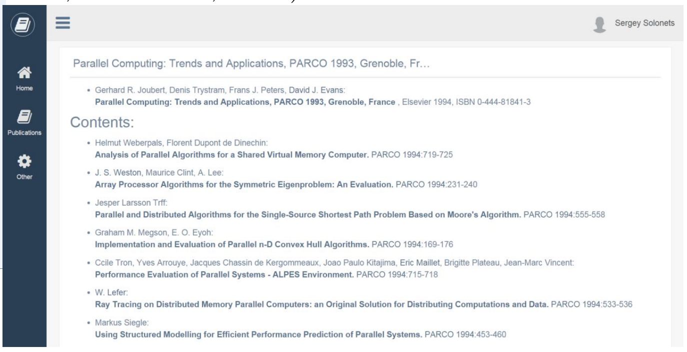
Incollection results

- Appa Rao Korukonda:
  Information Systems and Computer Science and Engineering. Wiley Encyclopedia of Computer Science and Engineering 2008:
  dx.doi.org
- Andrew D. McGettrick:
  Quality in Computer Science and Computer Engineering Education. Wiley Encyclopedia of Computer Science and Engineering 2008:
  dx.doi.org
- Allen B. Tucker, Peter Wegner:
  Computer Science and Engineering: The Discipline and Its Impact. The Computer Science and Engineering Handbook 1997: 1-15
- J. Mark Pullen, J. Goodlett McDaniel:
  Distance Education in Computer Science. Wiley Encyclopedia of Computer Science and Engineering 2008:
  dx.doi.org
- Edward K. Blum, Walter J. Savitch:
  The Software Side of Computer Science - Computer Programming. Computer Science, The Hardware, Software and Heart of It 2011: 53-68

2.2 Publications

.

Allows to search and view all information about all particular publication types(articles, inproceedings, proceedings, books, incollections, theses)



Parallel Computing: Trends and Applications, PARCO 1993, Grenoble, Fr...

- Gerhard R. Joubert, Denis Trystram, Frans J. Peters, David J. Evans:
  Parallel Computing: Trends and Applications, PARCO 1993, Grenoble, France , Elsevier 1994, ISBN 0-444-81841-3

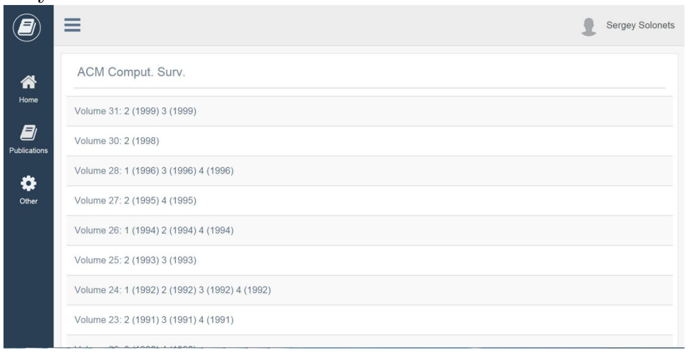
Contents:

- Helmut Weberpals, Florent Dupont de Dinechin:
  Analysis of Parallel Algorithms for a Shared Virtual Memory Computer. PARCO 1994:719-725
- J. S. Weston, Maurice Clint, A. Lee:
  Array Processor Algorithms for the Symmetric Eigenproblem: An Evaluation. PARCO 1994:231-240
- Jesper Larsson Trff:
  Parallel and Distributed Algorithms for the Single-Source Shortest Path Problem Based on Moore's Algorithm. PARCO 1994:555-558
- Graham M. Megson, E. O. Eyoh:
  Implementation and Evaluation of Parallel n-D Convex Hull Algorithms. PARCO 1994:169-176
- Ccile Tron, Yves Arrouye, Jacques Chassin de Kergommeaux, Joao Paulo Kitajima, Eric Maillet, Brigitte Plateau, Jean-Marc Vincent:
  Performance Evaluation of Parallel Systems - ALPES Environment. PARCO 1994:715-718
- W. Lefer:
  Ray Tracing on Distributed Memory Parallel Computers: an Original Solution for Distributing Computations and Data. PARCO 1994:533-536
- Markus Siegle:
  Using Structured Modelling for Efficient Performance Prediction of Parallel Systems. PARCO 1994:453-460

2.3 Other

2.3.1 *Journals*

Allows to navigate through journals and editions, which they consist from

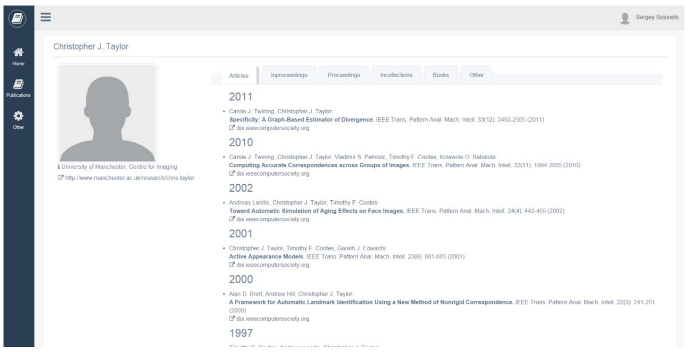


ACM Comput. Surv.

Volume 31: 2 (1999) 3 (1999)
Volume 30: 2 (1998)
Volume 28: 1 (1996) 3 (1996) 4 (1996)
Volume 27: 2 (1995) 4 (1995)
Volume 26: 1 (1994) 2 (1994) 4 (1994)
Volume 25: 2 (1993) 3 (1993)
Volume 24: 1 (1992) 2 (1992) 3 (1992) 4 (1992)
Volume 23: 2 (1991) 3 (1991) 4 (1991)

2.3.2 *Persons*

Allows to search and view information about persons and their activities


Sergey Solonets

Christopher J. Taylor

University of Manchester, Centre for Imaging

http://www.manchester.ac.uk/research/chris.taylor

Articles | Inproceedings | Proceedings | Incollections | Books | Other

2011

- Carole J. Twining, Christopher J. Taylor:
  Specificity: A Graph-Based Estimator of Divergence. IEEE Trans. Pattern Anal. Mach. Intell. 33(12): 2492-2505 (2011)
  doi.ieeecomputersociety.org

2010

- Carole J. Twining, Christopher J. Taylor, Vladimir S. Petrovic, Timothy F. Cootes, Kolawole O. Babalola:
  Computing Accurate Correspondences across Groups of Images. IEEE Trans. Pattern Anal. Mach. Intell. 32(11): 1994-2005 (2010)
  doi.ieeecomputersociety.org

2002

- Andreas Lanitis, Christopher J. Taylor, Timothy F. Cootes:
  Toward Automatic Simulation of Aging Effects on Face Images. IEEE Trans. Pattern Anal. Mach. Intell. 24(4): 442-455 (2002)
  doi.ieeecomputersociety.org

2001

- Christopher J. Taylor, Timothy F. Cootes, Gareth J. Edwards:
  Active Appearance Models. IEEE Trans. Pattern Anal. Mach. Intell. 23(6): 681-685 (2001)
  doi.ieeecomputersociety.org

2000

- Alan D. Brett, Andrew Hill, Christopher J. Taylor:
  A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence. IEEE Trans. Pattern Anal. Mach. Intell. 22(3): 241-251 (2000)
  doi.ieeecomputersociety.org

1997

You can see online demonstration on the http://mydblp.com.swtes
Warning: You need to register first

3. IMPLEMENTATION DESCRIPTION

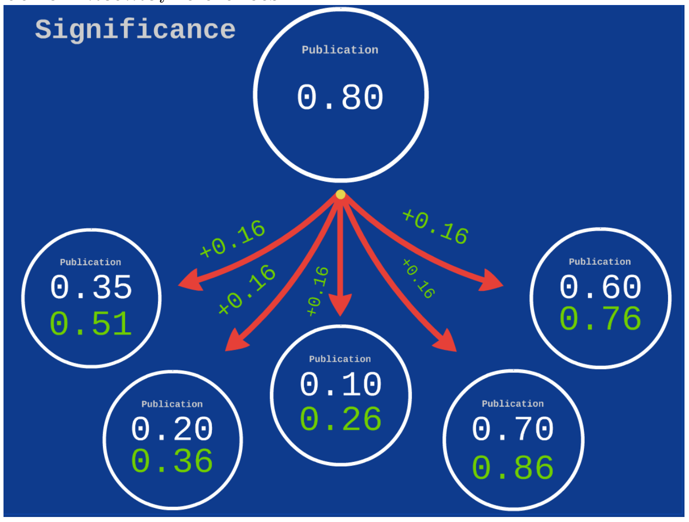
All implementation was done on **php** and **MySQL**, using framework **CodeIgniter**. As source was taken **DBLP**. Usage of xml file parsing and DBLP crawling together helped avoid a lot of uncoordinated data in xml file. Operations: Insertion, Update, Deletion implemented in crawler.We had to refuse from their realisation in web interface, because user actions in this case conflict with real-time crawler.

3.1 Sorting

Were implemented two sorting methods:

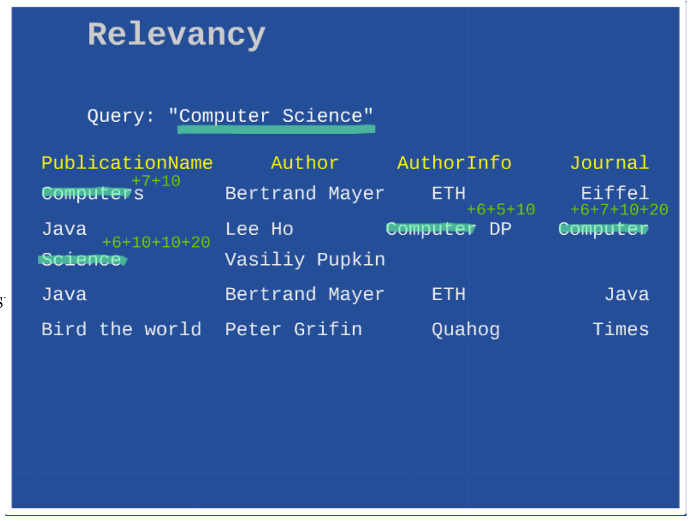
3.1.1 Reasonable first

Sorting publications using rank, which counting as a probability of random surfer to visit this publication by chaotically clicking on references on this site. For this purpose each person and publication getting initial rank and share it proportionally to all correlated entities(person or publication). This is a emulation of Google page rank algorithm. $Rank(Pub) = 0.5 + 0.85 * \sum_{i=1}^{n} \frac{Rank(Income_i)}{C(Income_i)}$, where $Income_i$ - i-th entity referred to this publication, $C(Income_i)$ - number of $Income_i$ references.



3.1.2 Relevant first

Each publication property getting their own weight. Query parse into words. Each searching publication getting weight as sum of weights(multiplied on word length) of attributes, in which these words were found. Also it is used raising coefficient in case, when word was found entirely, but not as a word part.



3.2 Searching

Were implemented two searching methods Quick and Deep. In both cases query is parsing into words. These word are searched in the specified columns. The difference is only about the columns:

3.2.1 Quick search

The set of columns was selected in such way, that all where clauses would push out all the heavy joins. Speed difference with Deep search in search performance using the same key word can reach 10 times.

3.2.2 Deep search

However Deep search is more convenient, because it searches through all publications information(even those are not displayed in description of this publication). For example, query "Innopolis" would display also publications, which were written in Innopolis.

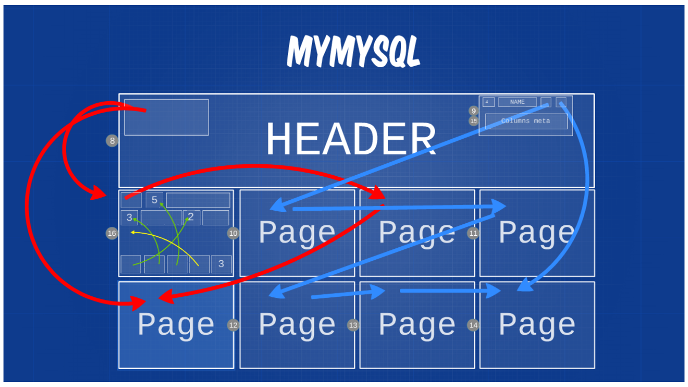
4. DEPLOYMENT

Phase 2 directory contains 2 directories: first one is applications, which is ready to be installed. Just copy it to your root server and edit configuration in **Application/Config**. Second one is routines contains parser, crawler and ranker(rank-crawler) See 2.1.1. MySql structure.sql also there. You can export data from http://vh229.spaceweb.ru/phpMyAdmin using login:solonets password:dblpdblpdblp

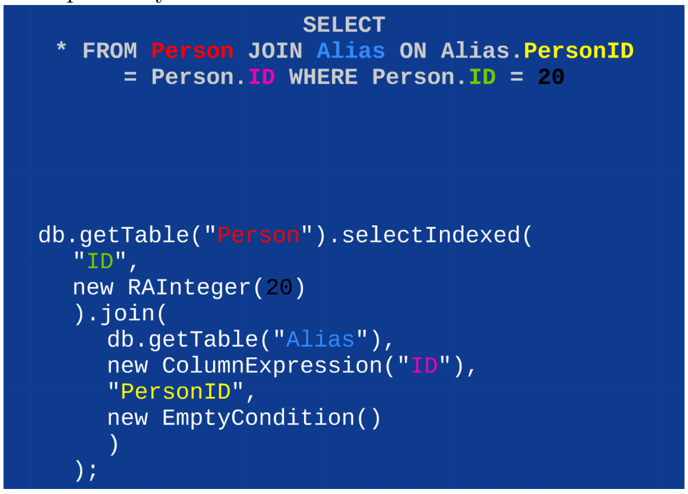
5. FILE SYSTEM

We stored all the data in a binary file. The first file page contains meta data, which is consists from:
4bytes(page size) | 4bytes(number of pages) | 4bytes(number of tables) | 4bytes[i-th table name length] | 4bytes[i-th table name] | 4bytes[i-th table start page] | 4bytes[i-th table current page] | 4bytes[i-th table last auto incemented] | 4bytes[i-th table number of attributes] | 4bytes[i-th table i-th attribute name length] | 4bytes[i-th table i-th attribute name] | 4bytes[i-th table i-th attribute type] | byte[i-th table i-th attribute auto incremented] | byte[i-th table i-th attribute indexed]. Let's consider each more closer.
page size - how much bytes was allocated per page

number of pages - how much pages contains this file
number of tables - how much tables contains this file
i-th table name length - name length of the i-th table
i-th table name - name of the i-th table
i-th table start page - from which page starts the records of i-th table
i-th table current page - on which page we will write next time
i-th table last auto incremented - last number used in auto incremented field of i-th table
i-th table number of attributes - number of attributes in i-th table
i-th table i-th attribute name length - i-th table i-th attribute name length
i-th table i-th attribute name - i-th table i-th attribute name
i-th table i-th attribute type - id of an i-th table i-th attribute type(it's int)
i-th table i-th attribute auto incremented - true/false
i-th table i-th attribute indexed - true/false
Each table records stored in their own pages. Page at the beginning contain pointer to the next page. At the end each page contains meta data about the page (pointers to records, last free space, free space quantity).



Indexing implemented by B+ tree. Also we havn't SQL parser, because in this case we would have to implement query optimisation. Using our implementation we can optimise queries by our selves.



```
SELECT
* FROM Person JOIN Alias ON Alias.PersonID
= Person.ID WHERE Person.ID = 20
```

```
db.getTable("Person").selectIndexed(
    "ID",
    new RAInteger(20)
    ).join(
        db.getTable("Alias"),
        new ColumnExpression("ID"),
        "PersonID",
        new EmptyCondition()
        )
    );
```